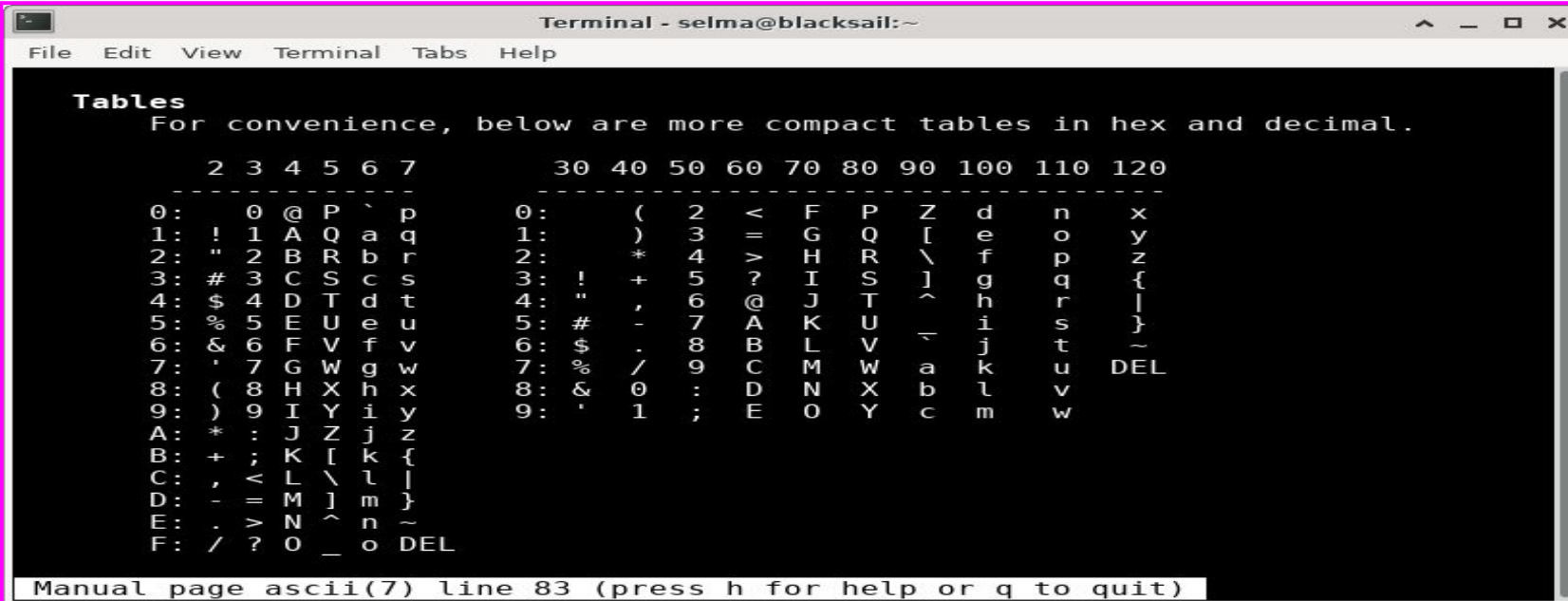


Représentation du texte

➡ L'ensemble des caractères ASCII.

Le code ASCII, “*American Standard Code for Information Interchange*” utilise 1 byte (1 octet) pour coder un caractère. Le bit de poids fort est 0. L'appellation internationale de l'ensemble de caractères/encodage ASCII est ISO 646-IRV, *International Reference version*.

La commande `man ascii` donne, en particulier, la liste des codes ascii pour 128 caractères et des tables pour les caractères imprimables.



```
Terminal - selma@blacksail:~
File Edit View Terminal Tabs Help

Tables
For convenience, below are more compact tables in hex and decimal.

    2 3 4 5 6 7          30 40 50 60 70 80 90 100 110 120
-----
0:  0 @ P ` p          0:  ( 2 < F P Z d n x
1:  ! 1 A Q a q          1:  ) 3 = G Q [ e o y
2:  " 2 B R b r          2:  * 4 > H R \ f p z
3:  # 3 C S c s          3:  ! + 5 ? I J S ] g q {
4:  $ 4 D T d t          4:  " , 6 @ K T U ^ h r |
5:  % 5 E U e u          5:  # - 7 A K U V ~ i s }
6:  & 6 F V f v          6:  $ . 8 B L V _ j t ~
7:  ' 7 G W g w          7:  % / 9 C M W a k l
8:  ( 8 H X h x          8:  & 0 : D N X b v
9:  ) 9 I Y i y          9:  ' 1 ; E O Y c m w
A:  * : J Z j z
B:  + ; K [ k {
C:  , < L \ l |
D:  - = M ] m }
E:  . > N ^ n ~
F:  / ? O _ o DEL

Manual page ascii(7) line 83 (press h for help or q to quit)
```

➡ Les ensembles de caractères ISO 8859.

Le code ASCII est largement suffisant pour représenter les caractères nécessaires à l'échange en langue anglaise. Afin d'inclure d'autres caractères comme les caractères accentués par exemple, le standard ISO/IEC 8859-1 est utilisé. C'est un code sur 8 bits, également appelé LATIN-1. Il coïncide sur sa partie basse avec le code ASCII.

ISO : International Organization for Standardization

<https://www.iso.org/home.html>

IEC : International Electrotechnical Commission

<https://www.iec.ch/>

ISO/IEC 8859 définit d'autres extensions à 8 bits du code ASCII notées ISO 8859-2,...,8859-16, chacune adaptée à une langue spécifique. Par exemple : ISO 8859-6 → Latin/Arabe.

La commande *man iso_8859_x* (x=1,...,16) donne la liste des codes sur 8 bits pour les caractères de l'ensemble de caractères ISO 8859-x au-delà du code ASCII.

➡ L'ensemble de caractères UCS.

UCS, The *Universal Character Set* maintenu par *UNICODE consortium* et ISO 10646 inclut un large nombre de caractères si bien que les caractères de n'importe quelle langue sont considérés. Il reste conforme aux ensembles de caractères ASCII et ISO 8859. UCS inclut également toutes catégories de caractères et de symboles comme les emojis.

<https://home.unicode.org/>

UCS est prévu pour un large espace de caractères. L'ensemble de caractères actuellement défini et utilisé est le standard Unicode.

A chaque caractère est attribué un nombre appelé point de code (*code point*) noté U+xxxxxx où ce qui suit le symbole + est en hexadécimal. Par exemple :

œ a pour point de code U+153 - 🕶️ a pour point de code U+1F60E

On peut trouver les points de code ici par exemple : <https://www.unicode.org/charts/>

ou utiliser des outils de recherche comme : <https://unicodelookup.com/>

➡ UTF-8 : un format de transformation pour l'Unicode.

Les points de code ont une longueur variable (actuellement 3 octets.) Comment les encoder ?

Sur une longueur fixe de 3 octets ?

Certainement pas car ce serait dramatique pour l'espace de stockage. Par exemple, un fichier ne contenant que des caractères ASCII occuperait le triple de la taille obtenue en gardant un simple encodage ASCII.

UTF-8, *Unicode Transformation Format* est prévu pour l'encodage de n'importe quel UCS caractère en utilisant jusqu'à 6 octets dans l'encodage du point de code correspondant. Faire *man utf-8* pour plus de détails.

Actuellement, le standard n'associe encore aucun caractère aux points de code au-delà de 0x10ffff si bien que tous les symboles et caractères considérés sont utf-8 encodables sur seulement 1,2,3 ou 4 octets.

➡ L'encodage UTF-8

RFC 3629

Un point de code est encodé par une séquence (binaire) de 1, 2, 3 ou 4 octets. Le premier octet de la séquence indique le nombre d'octets dans la séquence de la manière suivante :

Forme du premier octet de la séquence	Nombre d'octets dans la séquence
0xxxxxxx	1
110xxxxx	2
1110xxxx	3
11110xxx	4

➡ L'encodage UTF-8

RFC 3629

Les points de code U+0000 jusqu'à U+007f (US-ASCII characters repertoire) correspondent aux octets 00 jusqu'à 7f; soit les valeurs sur 7 bits du code ASCII. Ainsi, un texte ASCII est également une représentation UTF-8 valide.

Les points de code à partir de U+0080 utilisent au moins 2 octets dans leur encodage UTF-8. Les octets à partir du deuxième sont tous de la forme **10**xxxxxx où les 6 bits xxxxxx sont complétés à partir de la valeur du point de code.

C'est l'intervalle auquel appartient la valeur du point de code qui détermine le nombre d'octets dans la séquence UTF-8 correspondante selon la table donnée plus loin.

Exemple : U+5e9 → 0**101** **1110** **1001**. Son encodage UTF-8 est : **11010111** **10101001**

Ainsi, les bits x sont complétés à partir des bits dans la représentation du point de code, en commençant par le bit de poids faible de l'octet de poids faible et en remontant successivement vers les bits de plus en plus fort poids.

➡ L'encodage UTF-8

RFC 3629

La correspondance entre l'intervalle d'appartenance du point de code et le nombre d'octets dans la séquence UTF-8 qui le code est donnée par la table :

The table below summarizes the format of these different octet types.
The letter x indicates bits available for encoding bits of the character number.

Char. number range (hexadecimal)	UTF-8 octet sequence (binary)
0000 0000-0000 007F	0xxxxxxx
0000 0080-0000 07FF	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

L'exemple précédent U+5e9 illustre l'encodage UTF-8 d'un point de code qui appartient à l'intervalle correspondant à la deuxième ligne de cette table.